9. Mateo

Program Name: Mateo.java Test Input File: mateo.dat

Mateo has been studying polynomial functions in his algebra class and has no problem using the standard quadratic formula to find the roots, (or lack thereof) for a quadratic function. Higher-level polynomial functions are another matter when it comes to finding the roots because there is no simple formula. Mateo's computer science teacher explained that computational techniques could sometimes be used to find roots.

His teacher explained that if the sign of the function value changes from positive to negative or from negative to positive for two values of x, there is at least one root in that interval where $f(x) = 0$. However, when two values of x result in $f(x)$ being either positive or negative for both x values, it is unknown whether a root exists between those two values. When the values of $f(x)$ have opposite signs for a pair of x values, Mateo plans to select a new value for x between the two given values and then use that new value to shrink the interval in which the root exists until he closes in on a value where $f(x) = 0$.

Mateo decides to write a program to try the technique using the following polynomial function:

$$f(x) = 1.3x^4 - 35.1x^2 - 18.2x + 89.7$$

Assume at most one root will exist between each pair of x values. Realizing accuracy of floating-point values is an issue in programming, Mateo decides to use $f(x) = 0.0 \pm 0.0000001$ to find the root.

Input: An unknown number of lines and each line contains two different floating-point values for x, separated by whitespace, neither a root.

Output: When the pair of x values result in no sign change in $f(x)$, display "UNKNOWN"; otherwise, display the value of the root's x value with seven decimal places of accuracy.

Sample input:

```
-5.0 -3.0
0.0 -5.0
1.0 -4.0
-3.5 3.5
```

Sample output:

```
-4.5337926
UNKNOWN
-2.0804001
UNKNOWN
```